

### 31. Design Tools: Future Design Environments for Visualizing Building Performance

*Murray Milne*

Graduate School of Architecture and Urban Planning  
University of California  
Los Angeles, California, 90024-1467, USA

*In the future of Computer Aided Architectural Design (CAAD), architects clearly need more than just computer aided design and drafting systems (CAD). Unquestionably CAD systems continue to become increasingly powerful, but there is more to designing a good building than its three-dimensional existence, especially in the eyes of all the non-architects of the world: users, owners, contractors, regulators, environmentalists. . . .*

*The ultimate measure of a building's quality has something to do with how well it behaves over time. Predictions about its performance have many different dimensions; how much it costs to build, to operate, and to demolish; how comfortable it is; how effectively people can perform their functions in it; how much energy it uses or wastes. Every year dozens of building performance simulation programs are being written that can predict performance over time along any of these dimensions. That is why the need for both CAD systems and performance predictors can be taken for granted, and why instead it may be more interesting to speculate about the need for 'design tools'. A design tool can be defined as a piece of software that is easy and natural for architects to use, that easily accommodates three-dimensional representations of the building, and that predicts something useful about building's performance*

*There are at least five different components of design tools that will be needed for design environment of the future.*

#### **Introduction:**

It is easy to say that computers will profoundly change the way architects design buildings. The tough question is to figure out exactly what that future design environment be like. These environments will be powered, at least in part, by a never-ending series of CAD system up-dates and new releases, each one more wonderful than the last at describing buildings in three-space. All of these computerized drafting systems are the direct linear descendents of the tradition of representing buildings that began in ancient times with papyrus, then in modern times with linen and india ink.

Computers make it possible, however, to do something that is profoundly different from what architects could do before. Like having a Magic Lamp with a Genie inside, we have the ability to bring a design to life, run it forward in time, and watch how it behaves - not just what it looks like, but how warm or cold it feels, how bright or dark it looks, what it sounds like, and how well it accommodates human activities. This opens a new world of opportunities that for future design environments that we are just beginning to explore.

However, there is a cost to be paid from letting that Genie escape from the lamp. Architects in the future will undoubtedly face a perpetual series of increasingly stringent performance criteria, in large part because computers make it possible to predict performance with increasing precision. A useful analogy might be the continually increasing sophistication of medical diagnosis and treatment. Once the resource is available, patients insist on nothing less, as will our clients for design services in the future. The only difference is that the building sciences lag far behind most other areas of technological development.

Another example might be the ever tightening standards imposed on the automobile manufacturers for fuel efficiency and emissions reduction. Here legislation drives technology. The same is already true in architecture where new building codes, especially in the areas of life safety, energy, and the environment have "made a market" for all sorts of technological innovations that otherwise would languish because they were judged non-economic. Certainly in the future this trend will only continue, given the ever-expanding aspirations for public safety, and concern for the environment, global warming, and the consumption of energy. The fact that software is already available which can predict how our buildings will perform, makes it inevitable that those criteria will be imposed sooner or later. In short, it will happen because we can do it.

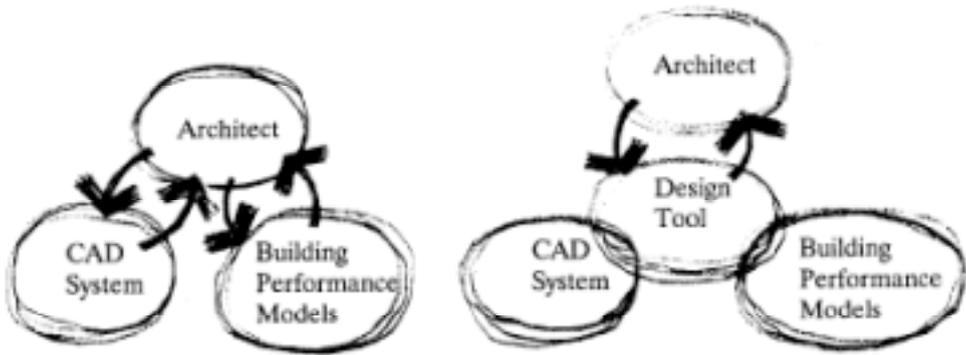
Many architects will see all this as a threat to their design freedom, but others will see it as an opportunity to increase our profession's relevance and value to society, and to expand our own intellectual horizons.

CAD systems, like AutoCad for example, are becoming increasingly sophisticated at manipulating shapes, forms, colors, and textures in three-space. While they can represent reflected ceiling plans and wall details, they have no direct way to manipulate the luminous, thermal, or auditory environment, or even structural performance or life safety today. We need completely separate simulation models to do that. Unfortunately we have no obvious way to link all of them together or to the CAD system. The interface of each piece of software is so idiosyncratic that no one (not even an architect) can master them all.

It can be argued that the process of designing a building involves the simultaneous manipulation of two distinctly different models; the geometric representation of the building in three-space, and a time-line representation of each of the different measures of its performance, such as its energy consumption, visual comfort, operating cost, etc. In the olden days before computers, the architect's task was to externalize the former by hand, and to create fantasies about the latter.

## **Design Tools**

What exactly is a design tool? It can be defined as a piece of computer software that has an easy-to-use interface, that allows the manipulation of the building's three-dimensional representation, and that shows the architect something useful about the performance of the building.



**Figure 1.** Future design environments will consist of at least four components, an Architect plus three pieces of software: a CAD system, a Building Performance Model, and at the center a Design Tool that allows them to communicate with each other.

Today, architects can use many different kinds of CAD systems (AutoCad, ComputerVision, etc), and many different kinds of performance models (BLAST, DOE-2, SUPERLITE, etc), but each has their own unique interface, and they do not talk to each other very well or share the same data. A design tool could sit between all three, acting as the interface for the architect to describe the building, then translating the building description to the performance model, and finally showing the architect how the building performs.

A design tool may incorporate a complex building performance model, or may simply act as an interface to it. Similarly it may incorporate its own building description database or else access the CAD system's database.

Architects of course might still try to communicate directly with their building simulation routines, but the availability of an easy-to-use design tool that is more efficient and friendly is a much more attractive option, and in fact today very few architects have ever touched an energy simulation, daylighting model, or cost analysis routine. In the future, the same might even be true of CAD systems, because although some architects today have learned how to operate them, to be honest they require a great deal of training in arcane manipulation and otherwise unproductive effort. The amount of practice time required to maintain competency or to reach high proficiency is beyond the resources of the most influential and highest paid architects in the firm. That is why the CAD system sits on the desks of the lowest paid entry-level drafters. Senior people will never embrace it until it is easier to use, has greater power, and is more portable than a felt-tip pen and a napkin.

Imagine as an alternative, a thin notebook-size tablet with a flat full-color display plus some kind of sonic pen that allows the architect to draw in a building directly in three-space, then with the touch of a "virtual button" rotate it, cast shadows throughout the day, and look at its passive solar performance throughout the year, then plug into a network and share it with everyone else.[1] When this 'fantasy' comes to pass, is there any doubt that sooner or later every chief designer will insist on using one?

## Specifications of a Design Tool for a Future Design Environment

There will always be something unique about how an architect designs a building, whether a computer helps out or not. The dialogue between a human and a CAD system engages the human in imagining a three-space model of the building, manipulating it cognitively, then describing pieces of it to the computer. Next, another rather different process takes place in which the architect looks at the computer's representation of the building and tries to evaluate it against various mentally stored criteria. Compared to computers, architects are incredibly sophisticated at the cognitive manipulation of building form, but are virtually incompetent at making precise calculations on complex models to predict building performance. Manipulating the data-rich world of building performance is an ideal task for computers.

A fully integrated design environment of the future is probably a long way off, but at UCLA over the past few years my graduate students and I have developed and tested many different components for such a system; some parts were carried only through a proof of concept, while others are fully developed and are in wide use.

The following five general categories describe most of the components that will be required in a Future Design Environment:

1. Establishment of initial design conditions;
2. Simplified three-dimensional data input;
3. Building performance predictions models;
4. Techniques for communicating complex data;
5. Automatic data conversion routines.

### 1. *Getting Off on the Right Foot: Initial Design Guidelines*

At the very beginning of the design process, the first few assumptions that the architect makes have a profound impact of the final design. These initial design decisions have far more influence than any other equal number of decisions made later in the process.

Consider for example, that at each different spot on planet Earth some ways of building will be more successful than others. Given the latitude and the hour-by-hour weather data, it is possible to say a great deal about the most appropriate design strategy. We have developed a program that reads the weather data for the site, then lists a set of design guidelines, rank ordered according to their relative importance.[2] There is nothing very complicated about this, in fact there is general agreement about what the guidelines are and when they apply. The big problem is how to translate them into architectural form. Here is an application that is begging for an expert system.

In another program we have built an expert system that needs to be told only four things about the project (location, building type, floor area, number of stories), then it will 'design' a simple energy conserving building.[3] It is not a very interesting building to look at, but when students use it as a basecase against which to test their own design, they find it is very hard to beat.

Dozens of other potential applications come to mind; topographic analysis, bubble diagrams, floor plan layout, code checks, perhaps automatically selected slide shows of comparable projects, or ultimately even a smart search through a CD-ROM of all the latest magazines.

## 2. *The Napkin: Quick Sketch as a Form of Non-verbal Inputs*

I wonder how many buildings have begun with a sketch on a paper napkin or the back of an envelope. The architect's earliest synthesis of relational and spatial issues needs to get instantly recorded and considered whole. Later versions become more literal and comprehensible, but the first sketch is just some squiggles and scratches that are incomprehensible to anyone who was not present at the conception. At this point the "building" is contradictory, ambiguous, and incomplete, certainly not something you would yet want to have to input into a solids modeler.

Imagine two or three architects around a drafting table, trying to solve a problem. The only thing an outsider would see is a series of incomprehensible sketches accompanied by grunting and head nodding, then suddenly one of them triumphantly draws another equally incomprehensible squiggle to which the others respond with relief, louder grunts, affirmative head nods, and now even spoken words will be heard like "yes...right...that's it."

Here is classic right brain activity, where information is being generated and shared without requiring intervening verbalization. Architects are especially skilled at this kind of non-verbal communication, especially as it relates to building form. It is one of our unique skills and one of the things that makes our profession so much fun. The kind of resonance and synchrony architects can feel with their co-designers, in the future must be part of their interactions with computers.

Something quite wonderful happens when the architect becomes completely immersed in the right-brain non-verbal world; there is magic in the speed and efficiency with which spatial concepts are generated and manipulated. Being forced to return to the verbal left-brain world is a jarring and disconcerting experience.

There is information in the act of drawing that computers could grasp; the weight of a line, the speed with which it is drawn, the sequence in which things are assembled. Imagine a computer program that knew if an X through a line meant to cut it in half here, or delete the whole line, or rotate everything about this point. Instead of a static drawing on a piece of paper, imagine a dynamic electronic image that could be stretched, rotated, flipped over, selectively erased, or de-constructed and re-formed in new ways. The real test of such a system would be if two architects who did not speak the same language, could work together on the same design. In fact, at UCLA we developed the prototype for such a system that can understand the architects gestures and modify the drawing accordingly (and it is now being marketed in Japan).[4]

Another problem is how to assign non-spatial attributes without stopping to use words and numbers. For every piece of the building the computer needs to know its R-value, reflectivity, thermal mass, weight, cost, presence of a vapor barrier, order of assembly, etc. Imagine a set of paint-pots that the architect could dip into and flood different surfaces with something like off-white stucco, standing seam copper roof, or terra-cotta Mexican floor tiles.[5] Imagine a box full of windows and doors that could be spotted onto the facade with a touch of the pen. The computer could of course render all of this in a spectacularly photo-realistic way, but more important is that now secretly attached to every surface is all the rest of the data needed to describe the building.

Most of the current CAD systems, although quite powerful, still do not have these capabilities. People who want to use them require a great deal of training (which is totally non-productive activity). Almost every action requires at least one translation between spatial

and verbal thinking which imposes an unnecessary cognitive load with its attendant inefficiencies and risks of error. Today's CAD systems virtually ignore non-spatial data, and in the end still only produce a frozen snapshot, having no way to bring the building to life and watch it run forward in time.

Thus the Design Environment of the Future needs some type of intuitively obvious way for an architect to communicate with a computer about how to describe three-dimensional building forms and see how they perform.

### 3. *Running the Clock Ahead: Building Performance Models*

Predicting the way a building might perform in the future is a fine art. Clearly there are many different ways to evaluate the success of a design; cost, energy, fame, number of design awards, etc. Obviously the degree of confidence of each of these different measures also varies widely.

Separate disciplines have sprung up around each of these building performance modeling areas. The oldest group is interested in simulating the structural behavior of buildings. But the fastest growing group is energy modelers; today there are dozens of different theoretical approaches and hundreds of different programs, all of which predict some aspect of how energy is used to heat and cool the building.

Three distinctions are worth pointing out: phase, focus, and speed. In each phase of the design process some kinds of energy models are appropriate while others are not. For instance, at the very beginning of the design process a model like BLAST or DOE-2 is useless; it requires the detailed input of every single building element, it takes a long time to execute, and produces stacks of tabular data. Instead what is needed is a model that works with a minimal "fuzzy" description of the building, executes in less time than it takes the architect's mind to wonder (probably about 15 seconds), and presents results in a way that can be instantly grasped by right-brain thinkers.[6] As the architect's design becomes more focused, the model must also incorporate the more detailed representation and produce increasingly precise results. At every phase in this process, the design tool must show the architect if the latest decisions are moving the project in the right direction.

Large detailed energy models come into their own toward the end of the design development phase, when the building description is essentially complete, and when there is sufficient time to run a series of parametric studies to 'fine-tune' details of the design.

Daylight modeling of buildings has recently become a topic of considerable interest, but here too the issues of phase, focus, and speed apply. At the phase of the process when the design of the facade is being articulated, a daylighting design tool can help answer questions about how such things as the size and placement of windows, and the dimensions of fins, overhangs, sills, and interior lightshelves on light distribution throughout interior spaces. When the details of the electric lighting are being integrated, more highly focused parametric performance studies can be run.[7] Computational techniques are known which will make it possible in the future to evaluate glare and visual comfort.

The final task for these models is code compliance.[8] But because building laws and energy codes test against minimum standards, therefore it is reasonable to expect that a building designed with good design tools should easily exceed these minimum standards.

#### 4. *Data-land: Envisioning Building Performance*

Sitting face to face at every terminal are two powerful information processors, a computer and an architect. Yet everything exchanged between them must pass through slow, low resolution devices that choke off precise complex communication.[9]

The computer may 'know' how the building will perform, but the most challenging task is to devise effective ways to share the computer's knowledge with the human. At UCLA we have been working on ways to display building performance over time. We have developed a 3-D plot that shows the state of the variable for every hour of the day for every month of the year.[10] The resulting shapes are easy to grasp at the highest level, and at the same time are rich with information at the most detailed level. The beauty and power of this approach is that it communicates in a direct right-brain mode, allowing the architect to pick up incredibly subtle distinctions that would otherwise be lost in a page full of numbers. Tufte defines information as the recognition of differences that make a difference. We have used this technique to display all kinds of different things about the building's performance; heat flow, illumination levels, cost of energy, ventilation rates, etc.

There are of course other ways to represent the performance of a building over time. For example, one could compute a kind of movie or video, and watch the way natural light moves through the room.[11] But Architects might also like to know if there is adequate light to perform a specific task, are there brightness ratios that would cause direct or reflected glare for any user, and what is the probability that users will find the space visually comfortable.[12] Imagine that the movie now shows a false-color representations of where and when any of these problems occur in the occupant's visual field; a checkerboard pattern shows where brightness ratios are too high, a splotch of purple shows the source of reflected glare.

#### 5. *Transformers: Data Conversion*

We all know myths about how wonderful the world becomes when we simply transform our problems into another domain. A popular Christmas gift recently was a stuffed frog. When you reached inside his mouth and pulled him inside-out he became (you guessed it) THE HANDSOME PRINCE. Here a simple and elegant translation from one data structure to another is a solution to the problem - unless you happened to be a lady frog. In the real world of CAD however, it is hard to find very many successful examples. We have a few examples of going from frogs to toads (DXF to TIF for instance), but getting all the way to a prince is another matter. Imagine how delighted energy consultants would be if buildings could be instantly translated from AutoCad to DOE-2 (DXF to BDL).[13] Then again maybe they wouldn't be delighted, because the possession of this little bit of secret knowledge is one of the reasons architects need energy consultants, but the lack of it is also one of the reasons why so few architects ever do an energy analysis in the first place. Would buildings be better if architects could immediately flip from one domain into the other instantly seeing an energy analysis, a cost comparison, an illumination level plot?

Thus another thing that a design tool must be able to do is not only to let the architect talk to the CAD system about the building, but also to let the performance model query the CAD system about the building. The CAD system knows a lot of the information that the performance model needs, but unfortunately not in the format that it needs. It is safe to assume that every CAD system and every performance model has its own different way to

store data, and it is not realistic to assume that any of them will be restructured simply to accommodate the needs of the other.

## Epilogue

All this has been only speculation what lies in the future for CAAD. But we all must admit that no matter how certain we are, we really never can be sure exactly how the future will turn out:

*The beautiful princess leaned down, and as her lips touched his green flesh there was a blinding flash . . . . .*

## References

1. The earliest and most fascinating speculations about future design environments came from Nicholas Negroponte. His books *The Architecture Machine* and *The Soft Architecture Machine* are still the richest and most stimulating collection of ideas about the future. Currently as head of MIT's Media Lab he is working with a world class team of software and hardware people to bring it all off. Negroponte's original specification for a sketch-book size workstation also included the requirement that it be waterproof so he could use it in the shower, and flexible so he could roll it up. Except for these last two (slightly dubious) requirements, all of the rest of his predictions have come to pass and are commercially available today. All that remains now is to reduce their size, weight, and cost (which inevitably happens to all computer hardware in any case).
2. Our climate analysis design tool has been through a long process of evolution at UCLA. A little bit of its history might give a sense of how long this kind of software development takes and how everyone builds on the work of those who have gone before. Our first computer graphic representation of NOAA Monthly Average climate data was by Bruce (Ted) Anderson in 1979. A few years later, Yun Kim carried this work a step further, developing an interactive program that students first used in the Building Climatology class. Next Wade Hokoda wrote a program to read weather tapes then to plot hourly values and monthly averages on the psychrometric chart that Baruch Givoni had developed in his book *Man Climate and Architecture* (Van Nostrand Reinhold), on which was also laid out the comfort zone and the zones showing the most appropriate passive solar building design strategies. His program kept track of the hourly values in each zone of the psychrometric chart and generated a list of passive design strategies rank-ordered according to the percentage of hours each was effective.

In his book *Energy Conservation Through Building Design* (McGraw-Hill), Don Watson included a chapter by Givoni and Milne that expanded on the technique of using the psychrometric chart as a design tool. When Watson and Ken Labs wrote their book *Climatic Design* (McGraw-Hill), they used this approach to identify the most appropriate passive design strategies for forty different cities, using a program written at Yale.

A micro-computer version of this design tool, called *Climate Consultant* was written at UCLA by Alex Acenas and Mark Clayton incorporating the list of fifty design guidelines that Watson and Labs developed in their book, it allows the architect to custom modify the climate "window" within which each guideline applies, in order to fine-tune the overall design strategy. Their program was not only friendly and colorful, but added new features like a *Timetable of Bioclimatic Needs* as developed by the Olgyay brothers in the 1940's. Currently this work is being pushed rapidly forward by Yung-Hsin Li who has added the capability of displaying wind data in a 3-D plot. He also automatically plots sun charts of the type Edward Mazria used in his book *The Passive Solar Energy Book*, except that he can overlay on these charts the *Bioclimatic Timetable* that showed the times when windows need to be shaded or to be exposed for direct gain. He is currently working on an expanded list of design guidelines and an expert system to aggregate them in more relevant ways.

3. Expert systems sometimes can be simple yet powerful, a very early example of which was written by Joel Loveland for *SOLAR-5*, to work at the very first moment of the design process. The architect only needs define four things; the building type, floor area, number of stories, and location of the nearest weather station. The program then creates a "building" complete with windows, sunshades, occupants, and lights.

*SOLAR-5* has a long history of development, beginning with the first version that we used in the *Building Climatology* class written by Shin Yoshikawa in 1978. Since then almost two dozen different Graduate Research Assistants have contributed to the project. The current version has all the attributes of a design tool as defined in this paper: the interface is friendly enough that beginning students are usually into their second or third design iteration within an hour after they first try it; the building can be described in a few minutes, it takes only a few minutes more for the energy model to calculate the buildings performance for all 24 hours and all 12 months; and finally the results are displayed graphically in a mode that portrays large quantities of complex data in a form easily grasped by architects or others trained in right-brain thinking.

4. The challenge of whether we could develop a completely non-verbal interface for architectural design problems was accepted by Yas Endo in his thesis of 1984. His CAD system watches the context, the sequence of prior events, to decide if the architect intends to enclose a form, erase a line, rotate a segment of the drawing, flip it over, etc. It can understand certain graphic input conventions like arrow heads and centerlines. At *Takanaka Architects* he has carried this work forward to the point where over a hundred systems have been marketed in Japan under the name of 'T-Board'.

5. At the University of Oregon, Charlie Brown suggested the paintpot metaphor as part of a Macintosh-based energy modeling system he was developing called Energy Scheming.

6. SOLAR-5 is being developed at UCLA to address these exact specifications of phase, focus, and speed.

7. A design tool called DAYLIT was developed at UCLA to help architects develop building facades by showing them the distribution of light through their space from any combination of windows, skylights, or clerestories. It has the added power that it can show how the building performs for all 24 hours of the day and all 12 months of the year. In addition, automatic or manually controlled lights can be added in three different zones of the room, and the resulting savings in building electric energy consumption can be shown graphically in the same way. This work was supported by a grant from Southern California Edison.

8. The California Energy Commission performs a valuable service for architects in the state by making available at almost no cost, microcomputer programs for evaluating whether a building complies with the energy code: MICRO-CHECK, CAL-RES, SCM, etc. The first versions of these programs required a huge amount of input data but gave back only one piece of output data: "complies" or "fails". More recent versions, however, are more like design tools, identifying parts of the building that are most problematic, allowing multiple alternatives to be evaluated, simplifying the input process, and producing graphic output.

9. Edward Tufte has just published an exquisite book entitled, *Envisioning Information* (Graphics Press), which he designed as a "celebration of the escape from flatland". He presents wonderful examples of techniques for representing complex data in elegant and easy-to-grasp ways, all of which will particularly appeal to right-brain thinkers. He describes a multitude of techniques that he loosely collects under headings like Micro/Macro Readings, Layering and Separation, Small Multiples (in which he discusses the attributes of three-dimensional plots like those produced by SOLAR-5 and DAYLIT), Color and Information, and Narratives of Space and Time. He is correct when he suggests that his book could serve as a catalog for the collection in the (hypothetical) Museum of Cognitive Art, it is an intellectual and aesthetic delight.

10. The three-dimensional plots produced by SOLAR-5 have proven to be exceptionally powerful in communicating complex data. A simple metaphor is helpful. Once students understand that the plot of a good passive solar building is saddle-shaped (maximum heat gain at mid-day in winter), and bad passive buildings are heat mountains (peak gain at mid-day in summer). When students see their first SOLAR-5 plot, you sometimes see them visibly stiffen and recoil, but within minutes they have grasped the concept and have become intellectually intrigued. Then you see them leaning closer to the terminal, gesturing knowingly, and speculating if design changes like rotating the building a few degrees will reduce a bump here and fill in a valley there. Finally when you see the CRT screen covered with fingerprints you can be sure that they fully internalized the concept.

11. The first design tool we developed at UCLA was SOLAR-2. It shows a kind of time lapse plot of the sequence of sunlight penetration patterns as they move through the room for any month of the year. The designer can then change the window shape, fins, overhang, etc., and immediately see the new set of annual patterns. At the same time in the background it is calculating things like radiation through the glazing, percentage of window exposed to full sun, etc. Another interesting graphic that this program can generate is an axonometric view of the room, which when taken from the point of view of the sun is particularly useful to the designer because, for example, if full shading is needed it shows exactly which piece of glass is exposed, or if maximum solar gain is required it shows if any part of a fin or overhang is obstructing the window. The current micro-computer version of SOLAR-2 was written by Rong Sheu, and is described in his thesis of 1986.

12. Currently there is a good deal of interest in programs that generate photo-realistic images, calculating and displaying the brightness of every surface in the room. The more complex programs like RADIANCE developed by Lawrence Berkeley Laboratory, require something like a weekend on a computer to generate an image. The SUPERLITE program, also by L.B.L. uses a more pragmatic approach of dividing all surfaces up into a relatively large grid, then calculating the radiant exchanges only at the centroid of each grid point. At UCLA we are currently in the midst of a joint research project with USC to develop a "user friendly" front end for SUPERLITE.

13. In fact at UCLA we have a number of systems that can make some impressive transformations between very different data structures. In 1986 Den Won Lin built an AutoLisp extension on AutoCad that would automatically generate a scheme description for SOLAR-5. If the Designer follows certain conventions like using reserved layers, points defined counter-clockwise, and figures that must be enclosed, then by simply hitting "energy analysis" on the pull-down menu a complete SOLAR-5 run would be executed and the results instantly displayed.

Another thesis just completed by Joy Chen makes the conversion from AutoCad to SUPERLITE. It requires only that the perimeter of the desired space be identified a special command, then all the three-space data would be written in the current format for SUPERLITE's .IN file. Upadi Yuliatmo is currently working under the joint USC/UCLA project to develop a translator between a generalized 3-D CAD system and the SUPERLITE .IN file.

## **Acknowledgements**

All of the graduate students mentioned in this paper worked as Research Assistants with financial support from the UCLA Academic Senate, The University Energy Research Group, and the U.S. Department of Energy. In many cases they also wrote their Masters Thesis on this work. SOLAR-2 was developed in cooperation with Robin Liggett at UCLA, and DAYLIT and the SUPERLITE input interface were developed in cooperation with Marc Schiler of USC. The authors appreciation to all is gratefully acknowledged.